

ARRAYS EN JAVASCRIPT



1. **concat()**: Combina dos o más arrays.
2. **join()**: Une todos los elementos de un array en una cadena y devuelve esta cadena.
3. **pop()**: Elimina el último elemento de un array y lo devuelve.
4. **push()**: Añade uno o más elementos al final de un array y devuelve la nueva longitud del array.
5. **shift()**: Elimina el primer elemento de un array y lo devuelve.
6. **unshift()**: Añade uno o más elementos al principio de un array y devuelve la nueva longitud del array.
7. **slice()**: Devuelve una copia de una parte del array dentro de un nuevo array.
8. **splice()**: Cambia el contenido de un array eliminando elementos existentes y/o añadiendo nuevos elementos.
9. **reverse()**: Invierte el orden de los elementos de un array.
10. **sort()**: Ordena los elementos de un array.
11. **map()**: Crea un nuevo array con los resultados de llamar a una función para cada elemento del array.
12. **filter()**: Crea un nuevo array con todos los elementos que pasan la prueba implementada por la función proporcionada.
13. **reduce()**: Aplica una función a un acumulador y a cada elemento de un array (de izquierda a derecha) para reducirlo a un único valor.
14. **forEach()**: Ejecuta una función dada una vez por cada elemento del array.
15. **indexOf()**: Devuelve el primer índice en el que se encuentra un elemento dado en el array, o -1 si no se encuentra.
16. **lastIndexOf()**: Devuelve el último índice en el que se encuentra un elemento dado en el array, o -1 si no se encuentra.
17. **includes()**: Determina si un array incluye un determinado elemento, devolviendo true o false según corresponda.

18. **every()**: Comprueba si todos los elementos de un array pasan una prueba proporcionada por una función.
19. **some()**: Comprueba si al menos un elemento de un array pasa una prueba proporcionada por una función.
20. **find()**: Devuelve el primer elemento de un array que cumple con la función de prueba proporcionada.
21. **findIndex()**: Devuelve el índice del primer elemento de un array que cumple con la función de prueba proporcionada.

Ejemplos:

- **reduce()**: Este método reduce un array a un solo valor aplicando una función a cada elemento del array.

```
<script>
  const numbers = [1, 2, 3, 4, 5];

  // Suma de todos los elementos del array
  const suma = numbers.reduce((accumulator, currentValue) => accumulator + currentValue, 0);
  console.log(suma); // Output: 15
</script>
```

- **filter()**: Este método crea un nuevo array con todos los elementos que pasan la prueba implementada por la función proporcionada.

```
</div>
<script>
  const numbers = [1, 2, 3, 4, 5];

  // Filtrar los números pares
  const numerosPares = numbers.filter(number => number % 2 === 0);
  console.log(numerosPares); // Output: [2, 4]
</script>
</body>
```